

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application: 2003年 3月24日

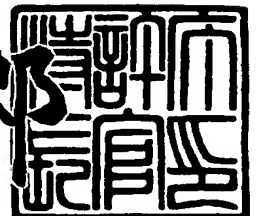
出願番号  
Application Number: 特願2003-079478  
[ST. 10/C]: [JP 2003-079478]

出願人  
Applicant(s): 株式会社東芝

2003年 7月 8日

特許庁長官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3054025

【書類名】 特許願

【整理番号】 13B0310901

【あて先】 特許庁長官殿

【国際特許分類】 H06L 9/10

【発明の名称】 プロセッサ、計算機及び優先度決定方法

【請求項の数】 17

【発明者】

    【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝  
研究開発センター内

    【氏名】 菅野 伸一

【発明者】

    【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝  
研究開発センター内

    【氏名】 宮本 幸昌

【特許出願人】

    【識別番号】 000003078

    【氏名又は名称】 株式会社 東芝

【代理人】

    【識別番号】 100083161

    【弁理士】

    【氏名又は名称】 外川 英明

    【電話番号】 (03)3457-2512

【手数料の表示】

    【予納台帳番号】 010261

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プロセッサ、計算機及び優先度決定方法

【特許請求の範囲】

【請求項 1】

あるデータ処理を少なくとも 1 以上に時分割した処理の一つに当たる、1 以上の実行単位が処理すべき処理データを記憶するための複数の記憶手段と、

ある実行単位に割り当てた前記複数の記憶手段の少なくとも 1 つからこの実行単位が処理すべきデータを受け取り、該実行単位に割り当てた前記複数の記憶手段の 1 つに該実行単位の実行結果を記憶するデータ処理手段と、

前記記憶手段のそれぞれから各記憶手段が記憶する処理データ量についての情報を受け取り、ある記憶手段が記憶する処理データ量から、該記憶手段をアクセスする実行単位の実行状況を求め、該実行状況から該実行単位の以降実行すべき順位を示す優先度を決定する優先度決定手段と、  
を備えたことを特徴とするプロセッサ。

【請求項 2】

前記優先度決定手段は、ある記憶手段が記憶する処理データ量がより多いほど、該記憶手段から処理すべきデータを受け取る実行単位に対し、より高い優先度を決定することを特徴とする、請求項 1 に記載のプロセッサ。

【請求項 3】

前記優先度決定手段は、ある記憶手段が記憶する処理データ量がより多いほど、該記憶手段に実行結果を記憶する実行単位に対し、より低い優先度を決定することを特徴とする、請求項 2 に記載のプロセッサ。

【請求項 4】

前記優先度決定手段は、ある記憶手段が記憶する処理データ量がより少ないほど、該記憶手段から処理すべきデータを受け取る実行単位に対し、より低い優先度を決定することを特徴とする、請求項 1 に記載のプロセッサ。

【請求項 5】

前記優先度決定手段は、ある記憶手段が記憶する処理データ量がより少ないほど、該記憶手段に実行結果を記憶する実行単位に対し、より高い優先度を決定す

ることを特徴とする、請求項 4 に記載のプロセッサ。

**【請求項 6】**

前記優先度決定手段は、ある実行単位に対し、該実行単位が処理すべきデータを受け取る記憶手段が記憶している処理データ量が同等であっても、該記憶手段が記憶する処理データ量が増加傾向にあるときと比較して、該記憶手段が記憶する処理データ量が減少傾向にあるときの方により高い優先度を決定することを特徴とする、請求項 2 及び請求項 4 に記載のプロセッサ。

**【請求項 7】**

前記優先度決定手段は、ある実行単位に対し、該実行単位が実行結果を記憶する記憶手段が記憶している処理データ量が同等であっても、該記憶手段が記憶する処理データ量が増加傾向にあるときと比較して、該記憶手段が記憶する処理データ量が減少傾向にあるときの方により低い優先度を決定することを特徴とする、請求項 3 及び請求項 5 に記載のプロセッサ。

**【請求項 8】**

前記優先度決定手段は、ある記憶手段が記憶する処理データ量が該記憶手段の記憶容量の上限を越えると判断したときは、該記憶手段から処理すべきデータを受け取る実行単位に対し、もっとも高位の優先度を決定することを特徴とする、請求項 1 に記載のプロセッサ。

**【請求項 9】**

前記優先度決定手段は、ある記憶手段が記憶する処理データ量が該記憶手段の記憶容量の上限を越えると判断したときは、該記憶手段に実行結果を記憶する実行単位に対し、もっとも低位の優先度を決定することを特徴とする、請求項 1 に記載のプロセッサ。

**【請求項 10】**

請求項 1 乃至請求項 9 のいずれかに記載のプロセッサを備えた計算機。

**【請求項 11】**

あるデータ処理を少なくとも 1 以上に時分割した処理の一つに当たる実行単位について、ある実行単位に割り当てた複数ある記憶装置の少なくとも 1 つから該実行単位が処理すべきデータを受け取り、

該実行単位に割り当てた前記複数の記憶装置の 1 つに該実行単位の実行結果を記憶し、

前記記憶装置のそれぞれから各記憶装置が記憶する処理データ量についての情報を受け取り、ある記憶装置が記憶する処理データ量から、該記憶装置をアクセスする実行単位の実行状況を求め、該実行状況から該実行単位の以降実行すべき順位を示す優先度を決定することを特徴とする優先度決定方法。

**【請求項 1 2】**

前記優先度を決定するときは、ある記憶装置が記憶する処理データ量がより多いほど、該記憶装置から処理すべきデータを受け取る実行単位に対しより高い優先度を決定し、

かつ記憶する処理データ量がより少ないほど低い優先度を決定することを特徴とする、請求項 1 1 に記載の優先度決定方法。

**【請求項 1 3】**

前記優先度を決定するときは、ある記憶装置が記憶する処理データ量がより多いほど、該記憶装置に実行結果を記憶する実行単位に対しより低い優先度を決定し、

かつ記憶する処理データ量がより少ないほど高い優先度を決定することを特徴とする、請求項 1 1 に記載の優先度決定方法。

**【請求項 1 4】**

前記優先度を決定するときは、ある実行単位に対し、該実行単位から処理すべきデータを受け取る記憶装置が記憶している処理データ量が同等であっても、該記憶装置が記憶する処理データ量が増加傾向にあるときと比較して、該記憶装置が記憶する処理データ量が減少傾向にあるときの方により高い優先度を決定することを特徴とする、請求項 1 2 に記載の優先度決定方法。

**【請求項 1 5】**

前記優先度を決定するときは、ある実行単位について、該実行単位が実行結果を記憶する記憶装置が記憶している処理データ量が同等であっても、該記憶装置が記憶する処理データ量が増加傾向にあるときと比較して、該記憶装置が記憶する処理データ量が減少傾向にあるときの方により低い優先度を決定することを特

徴とする、請求項 13 に記載の優先度決定方法。

**【請求項 16】**

前記優先度を決定するとき、ある記憶装置が記憶する処理データ量が該記憶装置の記憶容量の上限を越えると判断したときは、該記憶装置から処理すべきデータを受け取る実行単位に対し、もっとも高位の優先度を決定することを特徴とする、請求項 11 に記載の優先度決定方法。

**【請求項 17】**

前記優先度を決定するとき、ある記憶装置が記憶する処理データ量が該記憶装置の記憶容量の上限を越えると判断したときは、該記憶装置に実行結果を記憶する実行単位に対し、もっとも低位の優先度を決定することを特徴とする、請求項 11 に記載の優先度決定方法。

**【発明の詳細な説明】**

**【0001】**

**【発明の属する技術分野】**

本発明は、データフロー型プロセッサに関するものであり、特に入力段と出力段にバッファメモリを備えたプロセッサ及びこのプロセッサにおける実行スレッドの選択方法に関するものである。

**【0002】**

**【従来の技術】**

計算機等のプロセッサにデータを処理させる際にはより多くのデータ処理を効率よく行うために、複数のタスクを時分割に実行するマルチスレッド処理が一般的に採用されている（たとえば特許文献 1 を参照）。

**【0003】**

データ処理のためのマルチスレッド化を実現する方法がいくつか提案されている。その中にデータフローに従ってスレッドの実行順序を決定するデータフロー型マルチスレッド処理がある。このようなプロセッサでは、処理すべきデータの処理過程で必要とする処理を、その処理順序に従って実行する仕組みが取られている。このようにすると、処理過程にあたらないうスレッドは実行されることがなく、その時点で必要とする処理にプロセッサ能力を効率よく割り当てることが可

能となる。

【0004】

【特許文献1】

特開 2003-029984 公報

【0005】

【発明が解決しようとする課題】

しかしながら、各処理単位の処理後の出力結果に依存して次の実行スレッドを選択する従来のデータフロー型では、次のような場合に問題となる。

【0006】

たとえばリアルタイム性を要求する処理対象にあっては、処理の流れ（データフロー）に則さない処理が必要となる場合が多い。データフローから逸脱する処理は別途割り込み処理や作業状態の退避処理等のオーバヘッドを必要とする。また、このような割り込み処理はデータフローに乱れを生じさせ、処理効率を低下させてしまう。

【0007】

本発明はこのような問題に鑑みてなされたものであり、リアルタイム性の損失を最小限に抑えることが可能なデータフロー型プロセッサ、このプロセッサを利用した計算機及び優先度決定方法を提供することを目的とする。

【0008】

【課題を解決するための手段】

本発明によれば、

あるデータ処理を少なくとも1以上に時分割した処理の一つに当たる、1以上の実行単位が処理すべき処理データを記憶するための複数の記憶手段と、

ある実行単位に割り当てた前記複数の記憶手段の少なくとも1つからこの実行単位が処理すべきデータを受け取り、該実行単位に割り当てた前記複数の記憶手段の1つに該実行単位の実行結果を記憶するデータ処理手段と、

前記記憶手段のそれぞれから各記憶手段が記憶する処理データ量についての情報を受け取り、ある記憶手段が記憶する処理データ量から、該記憶手段をアクセスする実行単位の実行状況を求め、該実行状況から該実行単位の以降実行すべき



順位を示す優先度を決定する優先度決定手段と、  
を備えたことを特徴とするプロセッサが提供される。

【0009】

及び上記プロセッサを備えた計算機が提供される。

【0010】

また、

あるデータ処理を少なくとも1以上に時分割した処理の一つに当たる実行単位について、ある実行単位に割り当てた複数ある記憶装置の少なくとも1つから該実行単位が処理すべきデータを受け取り、  
該実行単位に割り当てた前記複数の記憶装置の1つに該実行単位の実行結果を記憶し、

前記記憶装置のそれぞれから各記憶装置が記憶する処理データ量についての情報を受け取り、ある記憶装置が記憶する処理データ量から、該記憶装置をアクセスする実行単位の実行状況を求め、該実行状況から該実行単位の以降実行すべき順位を示す優先度を決定することを特徴とする優先度決定方法が提供される。

【0011】

【発明の実施の形態】

本発明の実施形態について、図を用いながら説明する。

【0012】

(第1の実施形態)

図1は第1の実施形態におけるプロセッサのブロック図の一例である。図1には、複数のFIFO(First-In First-Out：先入先出型メモリ)101、選択部102、実行制御部103、キャッシュ104、切替部105、複数のレジスタセット106、メモリバス107、記憶部108及び演算処理部109が示されている。

【0013】

FIFO101はメモリから構成されており、先に記憶したものから先に読み出すことが可能な記憶装置である。それぞれのFIFO101は1からn(nは任意)の複数備えている。それぞれのFIFO101の記憶容量は必ずしも同じである必要は

無いが、複数のスレッドを平等に処理するためには同一の構成である方が好ましい。

#### 【0014】

ここでいうスレッドとは、プロセッサが実行すべき一つの処理を複数のある時間間隔に分割して割り当てた後の処理単位またはその処理単位の実行を指す。スレッドは実行制御部103が行う各部の制御と、この制御下で、予め与えられた所定のプログラムコードを演算処理部109で実行することで実現する。本実施形態のプロセッサはデータフロー型なので、分割した複数のスレッドを1つの処理データに対して順次処理してゆくことで一つの処理が完了する。スレッドは並列処理されるようになっており、非同期に複数の処理データのデータフロー型処理を行うことができる。このように一つの処理をスレッド化するのは、主にリアルタイムOSとったマルチスレッド処理を実現するために行われている。

#### 【0015】

選択部102は複数備えたFIFO101の中から実行制御部103が指定するFIFOを選択する機能を備えている。ここで選択したFIFO101は演算処理部109によって読み書きが成される。

#### 【0016】

実行制御部103は、本実施形態のプロセッサが行う処理全般を制御する機能を備えている。ここには、どのスレッドを優先的に実行するかを示す優先度を保持するテーブルも含まれる。実行制御部103はFIFO101などの情報をもとにこのテーブルを設定し、テーブルの内容にしたがってどの処理データを処理するのどのスレッドの実行を行うかを各部に指示する機能を有する。

#### 【0017】

キャッシュ104は演算処理部109がスレッドに対応するプログラムコードを実行する際に、読み書き速度の遅い記憶装置へ逐次アクセスすることを回避するために設けられている。一般に演算処理部109の処理速度と比較して、メインメモリや磁気ディスク装置といった大容量記憶装置のアクセス速度は遅い。キャッシュ104は大容量記憶装置程の記憶容量は無いものの、アクセス速度の速い記憶装置を用いている。頻繁にアクセスするデータおよびプログラムコードに

については一旦キャッシュ104に記憶することで読み書きの時に演算処理部109が待たされるのをなくし、できる限り演算処理部109の処理量を稼ぐために用いられる。

#### 【0018】

切替部105は複数のレジスタセット106の内から処理に必要な1つ以上のレジスタセットをアクセス可能に選択するものである。演算処理部109は、この切替部105を介して処理に必要なデータが格納されているレジスタセットをアクセスする。

#### 【0019】

レジスタセット106は実行制御部103がプログラムコードの実行に必要な各種レジスタ（一時記憶装置）をまとめたものである。レジスタセット106には、たとえば計算用レジスタ、アドレッシング用レジスタ、スタックポインタといったものが含まれている。本実施形態ではこのレジスタセット106が1からm（mは任意）の複数備えられている。それぞれのレジスタセット106は、必ずしも同じ構成である必要は無いが、それぞれのスレッドがどのレジスタセットを使用しても処理できるためには同一の構成であることが好ましい。

#### 【0020】

メモリバス107は、キャッシュ104、複数のレジスタセット107及び記憶部108の間でデータの授受が行えるように設けられたものである。それぞれに記憶されたデータは、メモリバス107を通じてデータの移動が行える。データの移動は実行制御部103があたる。

#### 【0021】

記憶部108は、本実施形態のプロセッサが処理のために実行するプログラムコードや、処理の対象となる処理データを記憶する。場合によってはキャッシュ104及びレジスタセット107に記憶したデータの一時退避場所として使用される。。

#### 【0022】

演算処理部109は、記憶部108またはキャッシュ104が記憶するプログラムコードを実行する機能を有する。プログラムコードの実行に際しては、実行

制御部 103 の指示によりどの FIFO 101、どのレジスタセット 106 を使用して、どのスレッドを処理するのかを決定する。

#### 【0023】

図 2 に本実施形態におけるプロセッサのスレッド処理の概要を示す。

本実施形態のプロセッサはデータフロー型であるから、処理データは入力のもたらされ、基本的に一つの道筋（フロー）を経て出力が得られる。入力された処理データは、演算処理部 109 が予め記憶部 108 に記憶したプログラムコードに従って逐次実行する処理（各スレッド 201）を行い、出力結果として出力される。次段の処理がある場合には、出力された出力結果は次段の入力データとなる。

#### 【0024】

入力データである処理データが入力されると、実行制御部 103 が指示した複数ある FIFO 101 の一つである FIFO 101-A がこれを記憶する。FIFO 101-A は実行制御部 103 などからの求めに応じて、あるいは自発的に FIFO 101-A 内の FIFO 蓄積量を実行制御部 103 に対して状態報告 202-A として通知する。

#### 【0025】

実行制御部 103 の指示により演算処理部 109 がスレッド 201-1 を実行すると、FIFO 101-A 内のデータを入力データとした処理を開始する。このとき FIFO 101-A 内のデータは、先に記憶したものから順に読み出される。このスレッドが行った処理結果は実行制御部 103 の指示により FIFO 101-B に記憶される。FIFO 101-B もまた FIFO 101-A と同様に、実行制御部 103 に対して FIFO 蓄積量を含む状態報告 202-B を通知する。

#### 【0026】

上記の処理がスレッド 201-2 からスレッド 201-x（x は任意）まで処理を終えると、処理結果が出力として出力される。

#### 【0027】

演算処理部 109 がどのスレッドを実行状態とするかは、実行制御部 103 が実行指示 203 によって指示している。実行制御部 103 は、FIFO 101-A か

らFIFO101-nが通知する状態報告202-Aから状態報告202-nの情報をもとに自らが備える実行優先度テーブルに優先度を設定し、この優先度から実行指示203を決定する。実行指示203の決定に際しては、複数の状態報告202のうち一部だけを使用して決定した優先度に従っても良いし、すべての状態報告202を考慮して決定した優先度から最優先のスレッド201を決定してもよい。好ましくはすべての状態報告202からそれぞれのスレッド201の実行順位を決定するようにし、処理全体の効率化を図るようにした方がよい。

#### 【0028】

以降、実行制御部103が上記スレッドの実行順位を決定するための方法について説明する。

#### 【0029】

図3に本実施形態におけるプロセッサの実行制御部103が、あるスレッドについての入力側FIFO蓄積量に対するスレッドの実行優先度を決定する方法の一例を示す。図3に示す縦軸に当たるFIFO蓄積量は入力側FIFOであるから、たとえば図2に示すところのスレッド201-1に対するFIFO101-A、スレッド201-2に対するFIFO101-Bとなる。

#### 【0030】

あるスレッドの入力側FIFOに処理待ちデータが多く蓄積されているということは、すなわちそのスレッドの処理が遅れていることを意味する。よってそのスレッドの優先度を上げ、処理待ちデータを速やかに処理する必要がある。このため図3に示すようにFIFO蓄積量が増えるに従って実行され易くなる（優先度が高くなる）よう設定する。交点302に示すように、FIFO蓄積量が高くなるほど実行制御部103が保持する各スレッドの優先度を示すテーブルにより高い優先度が設定される。

#### 【0031】

図3に示す例ではFIFO蓄積量とスレッド実行優先度の関係を比例直線301で示しているが、必ずしも直線である必要はない。たとえばFIFO蓄積量が上限に近づくに連れて、優先度の上がり方が大きくなるようにした比例曲線303であっても良い。この場合にはFIFOが溢れることを効果的に防止することができる。ま

た、曲線や直線に依らずプロセッサの設計の上で実装が簡易となるように優先度が段階的あるいは階段状に高くなるようにしても良い。一例として、FIFO蓄積量にあるしきい値を設け、このしきい値を越えるまではスレッドの実行をしない、またはしきい値を切るまでは実行しつづけるといった操作も可能である。

#### 【0032】

図4は本実施形態におけるプロセッサの、あるスレッドについて出力側FIFO蓄積量に対するスレッドの優先度決定方法の一例を示す。図4に示す縦軸に当たるFIFO蓄積量は出力側FIFOであるから、たとえば図2に示すところのスレッド201-1に対するFIFO101-B、スレッド201-2に対するFIFO101-Cとなる。

#### 【0033】

あるスレッドの出力側FIFOに処理待ちデータが多く蓄積されているということは、すなわちその後続のスレッドの処理が遅れていることを意味する。このままスレッドを実行すると、出力側のFIFOが溢れてしまう可能性がある。そこでそのスレッドの優先度を下げ、出力側のFIFOが溢れないように処理を抑制する必要がある。このため図4に示すように出力側のFIFO蓄積量が増えるに従って実行され難くなる（優先度が低くなる）よう設定する。交点402に示すように、FIFO蓄積量が高くなるほど実行制御部103が保持する各スレッドの優先度を示すテーブルにより低い優先度が設定される。

#### 【0034】

図4に示す例ではFIFO蓄積量とスレッド実行優先度の関係を比例直線401で示しているが、必ずしも直線である必要はない。たとえば出力側のFIFO蓄積量が上限に近づくに連れて、優先度の下がり方が大きくなるようにした比例曲線403であっても良い。この場合には出力側FIFOが溢れることを効果的に防止することができる。また、曲線や直線に依らずプロセッサの設計の上で実装が簡易となるように優先度が段階的あるいは階段状に低くなるようにしても良い。一例として、FIFO蓄積量にあるしきい値を設け、このしきい値を切るまではスレッドの実行をしない、またはしきい値を越えるまで実行しつづけるといった操作も可能である。

**【0035】**

上記したように実行制御部103は入力側FIFOから得られた優先度と出力側FIFOから得られた優先度の両方を総合した優先度、あるいは入力側FIFOに基づく優先度に従って、スレッドの優先度を決定する。

**【0036】**

このとき入力側FIFO蓄積量から得られた優先度と出力側FIFO蓄積量から得られた優先度とが相反する優先度を示す場合も考えられる。たとえば入力側出力側の両方が蓄積量の上限に近い場合が考えられる。この場合には出力側のFIFO蓄積量の空き領域に、入力側FIFOに記憶された処理データを処理した後の出力結果が格納できるだけの領域があることを条件に、入力側FIFO蓄積量に基づく優先度を優先する。この場合を除けば、出力側FIFOが溢れるのを防止するため出力側FIFO蓄積量に基づく優先度を優先する、あるいはこのスレッドの実行を一時的に禁止する。

**【0037】**

図5に本実施形態におけるプロセッサの、次に実行するスレッドを決定するためのフローの一例を示す。

**【0038】**

まず、全FIFOの蓄積量を取得する(S1)。取得の方法については、各FIFOに問い合わせても良いし、各FIFOから自主的に報告されたものであっても構わない。取得の方法については限定しない。

**【0039】**

次に、各FIFOについて上限の蓄積量を超えそうなFIFOがあるかどうかを検査する(S2)。検査の結果超えそうなFIFOがある場合には、そのFIFOに関連するスレッドについて緊急処理を行う(S3)。

**【0040】**

緊急処理には、たとえば次のような処理がある。一つには該当するFIFOを入力側に持つスレッドを割り込みにより最優先で処理させるものである。割り込みとは通常の処理の順序を強制的に変更させる手段を言う。他のスレッドに優先して処理することにより、当該FIFOの蓄積量を減らすことができる。もう一つは該当

するFIFOを出力側に持つスレッドの実行を一時的に禁止するものである。当該スレッドの実行を禁止することで、当該FIFOに追加されるデータを止め、蓄積量に余裕ができるようにする。

#### 【0041】

緊急処理の内容は処理状況や処理データ特性により柔軟に決定すべきものであり、上記した例に限定されるものではない。

#### 【0042】

オーバフローしそうなFIFOが無いときは、次に全スレッドの優先度を求めたかどうかを判断する (S4)。

#### 【0043】

優先度を求めたスレッドがすべてでない場合は、処理中あるいは待機中のあるスレッドの一つを選択する (S5)。そして選択したスレッドが使用する入力側および出力側FIFOの蓄積量から、上述した方法で当該スレッドの優先度を決定する (S6)。決定後、再び全スレッドの優先度を求めたかを判断する (S4)。

#### 【0044】

すべてのスレッドについて優先度を求めたと判断すると、優先度を求めたスレッドの中からもっとも高位の優先度を持つスレッドを選択する (S7)。

#### 【0045】

上記のようにすると処理中あるいは待機中のスレッドから、次に実行すべきスレッドを選択することができる。

#### 【0046】

上記したように構成すると、データフロー型プロセッサの処理能力を効率よく必要とする処理に割り当てられるとともに、入力側及び出力側のFIFO蓄積量に応じて実行すべきスレッドを選択することが可能となる。

#### 【0047】

また本実施形態におけるプロセッサを搭載した、データフロー型の処理を行う計算機とすることも可能である。

#### 【0048】

(第2の実施形態)



第2の実施形態におけるプロセッサのブロック図の一例を示す図、およびスレッド処理の概要を示す図は、第1の実施形態における図1および図2と同様である。

#### 【0049】

図6に本実施形態におけるプロセッサの、あるスレッドについての入力側FIFO蓄積量に対するスレッドの優先度決定方法の一例を示す。図6に示す縦軸に当たるFIFO蓄積量は入力側FIFOであるから、たとえば図2に示すところのスレッド201-1に対するFIFO101-A、スレッド201-2に対するFIFO101-Bとなる。

#### 【0050】

あるスレッドの入力側FIFOに処理待ちデータが多く蓄積されているということは、すなわちそのスレッドの処理が遅れていることを意味する。よってそのスレッドの優先度を上げ、処理待ちデータを速やかに処理する必要がある。このため図6に示すようにFIFO蓄積量が増えるに従って実行され易くなる（優先度が高くなる）よう設定する。

#### 【0051】

第1の実施形態との相違は、FIFO蓄積量の増加時と減少時で異なるスレッド実行優先度とするところである。同じFIFO蓄積量のところを見ても、たとえばFIFO蓄積量が増加する際の交点601に当たる優先度は、FIFO蓄積量が減少する際の交点602に当たる優先度に比べて低くなる。

#### 【0052】

入力側FIFOの蓄積量とスレッド処理との関係を見たとき、スレッドが実行されれば入力側FIFOの蓄積量が減ってゆき、逆に実行されなければ増加する傾向にある。このとき交点602があるFIFO蓄積量の減少時について考えると、入力側FIFOの蓄積量が減っていても、そのスレッドの優先度の下がり方は緩やかになっている。逆に交点601があるFIFO蓄積量の増加時について考えると、入力側FIFOの蓄積量が増えていても、そのスレッドの優先度の上がり方は緩やかである。つまり入力側FIFOの蓄積量が変化しても、一旦処理を始めたスレッドは優先度が高いまま維持されることから連続して実行される可能性が高い。逆に優先度が

低くなり実行され難くなったスレッドは、入力側FIFOの蓄積量がより高くなりな  
いと実行され難いことを意味している。

#### 【0053】

図1に示したキャッシュ104は、前述したようにスレッド処理に必要とする  
処理データを記憶容量は少ないものの高速にアクセスできるように設けられてい  
る。記憶容量が少ないことから、一旦読み込んだあるスレッドの処理データは他  
のスレッドが実行されると上書きされて消滅する可能性が高い。消滅した処理デ  
ータは再びアクセス速度の遅い記憶装置からキャッシュ104上に読み込まねば  
ならない。入力側FIFOの蓄積量の変動により実行されるスレッドが頻繁に入れ替  
わると、必然的に他の記憶装置からキャッシュ104に読み込まねばならないデ  
ータの量及び回数が増える。

#### 【0054】

図1に示したレジスタセット106についても同様に、実行中及び待機中のス  
レッドがレジスタセット106の数を上回る場合には、他の記憶装置に一時的に  
退避する必要がある。これはある時点にレジスタセット106に割り当てられな  
かったスレッドを実行するために、他のスレッドが占有していたレジスタセット  
106をこのスレッドのために開放する必要があるからである。レジスタセット  
106の内容を記憶装置間で移動しなければならないことから、頻繁なスレッド  
の切替は、キャッシュ104と同様にオーバヘッドを発生させる。

#### 【0055】

本実施形態のプロセッサではあるスレッドが頻繁に実行されやすい状態を実現  
し、キャッシュ104へのデータ読み込みのオーバヘッドを削減する。よってプ  
ロセッサの持つ処理能力を、本来のデータ処理に振り向けることが可能となる。

#### 【0056】

図6に示す例ではFIFO蓄積量とスレッド優先度の関係を楕円で示しているが、  
必ずしも楕円である必要はない。たとえばプロセッサの設計の上で実装が簡易と  
なるように優先度が段階的あるいは階段状に高くなるようにしても良い。

#### 【0057】

図7は本実施形態におけるプロセッサの、あるスレッドにについて出力側FIFO

蓄積量に対するスレッドの優先度決定方法の一例を示す。図 7 に示す縦軸に当たる FIFO 蓄積量は出力側 FIFO であるから、たとえば図 2 に示すところのスレッド 201-1 に対する FIFO 101-B、スレッド 201-2 に対する FIFO 101-C となる。

#### 【0058】

あるスレッドの出力側 FIFO に処理待ちデータが多く蓄積されているということは、すなわちその後続のスレッドの処理が遅れていることを意味する。このままスレッドを実行すると、出力側の FIFO が溢れてしまう可能性がある。そこでそのスレッドの優先度を下げ、出力側の FIFO が溢れないように処理を抑制する必要がある。このため図 7 に示すように出力側の FIFO 蓄積量が増えるに従って実行され難くなる（優先度が低くなる）よう設定する。

#### 【0059】

第 1 の実施形態との相違は、FIFO 蓄積量の増加時と減少時で異なるスレッド実行優先度とするところである。同じ FIFO 蓄積量のところを見ても、たとえば FIFO 蓄積量が増加する際の交点 702 に当たる優先度は、FIFO 蓄積量が減少する際の交点 701 に当たる優先度に比べて高くなる。

#### 【0060】

出力側 FIFO の蓄積量とスレッド処理との関係を見たとき、スレッドが実行されれば出力側 FIFO の蓄積量が増えてゆき、逆に実行されなければ減少する傾向にある。このとき交点 702 がある FIFO 蓄積量の増加時について考えると、出力側 FIFO の蓄積量が増えていっても、そのスレッドの優先度の下がり方は緩やかになっている。逆に交点 701 がある FIFO 蓄積量の減少時について考えると、出力側 FIFO の蓄積量が減っていても、そのスレッドの優先度の上がり方は緩やかである。つまり出力側 FIFO の蓄積量に変化しても、一旦処理を始めたスレッドは優先度が高いまま維持されることから連続して実行される可能性が高い。逆に優先度が低くなり実行され難くなったスレッドは、出力側 FIFO の蓄積量がより低くならないと実行され難いことを意味している。

#### 【0061】

この効果については、上述した図 6 の説明について参照されたい。

**【0062】**

図7に示す例ではFIFO蓄積量とスレッド実行優先度の関係を楕円で示しているが、必ずしも楕円である必要はない。たとえばプロセッサの設計の上で実装が簡易となるように優先度が段階的あるいは階段状に高くなるようにしても良い。

**【0063】**

上記したように実行制御部103は入力側FIFOから得られた優先度と出力側FIFOから得られた優先度の両方を総合した優先度、あるいは入力側FIFOに基づく優先度に従って、スレッドの優先度を決定する。

**【0064】**

このとき入力側FIFO蓄積量から得られた優先度と出力側FIFO蓄積量から得られた優先度とが相反する優先度を示す場合も考えられる。たとえば入力側FIFO及び出力側FIFOの両方に処理データが蓄積されていないときである。この場合は優先的にスレッドの実行をする必要はないので入力側FIFO蓄積量に基づく優先度を優先する。

**【0065】**

一方、入力側出力側の両方が蓄積量の上限に近い場合も考えられる。この場合には出力側のFIFO蓄積量の空き領域に、入力側FIFOに記憶された処理データを処理した後の出力結果が格納できるだけの領域があることを条件に、入力側FIFO蓄積量に基づく優先度を優先する。この場合を除けば、出力側FIFOが溢れるのを防止するため出力側FIFO蓄積量に基づく優先度を優先する、あるいはこのスレッドの実行を一時的に禁止する。

**【0066】**

本実施形態におけるプロセッサの、次に実行するスレッドを決定するためのフローの一例を示す図は、第1の実施形態における図5と同様である。

**【0067】**

なお、本発明は上記実施形態そのままに限定されるものではなく、実施段階ではその要旨を逸脱しない範囲で構成要素を変形して具体化できる。また、上記実施形態に開示されている複数の構成要素の適宜な組み合わせにより、種々の発明を形成できる。例えば、実施形態に示される全構成要素から幾つかの構成要素を

削除してもよい。さらに、異なる実施形態にわたる構成要素を適宜組み合わせてもよい。

【0068】

【発明の効果】

本発明によれば、リアルタイム性の損失を最小限に抑えることが可能なデータフロー型プロセッサ、このプロセッサを利用した計算機及び優先度決定方法を提供することができる。

【図面の簡単な説明】

【図1】

本発明の第1の実施形態にかかるプロセッサのブロック構成の一例を示す図である。

【図2】

本発明の第1の実施形態にかかるプロセッサのスレッド処理の概要を示す図である。

【図3】

本発明の第1の実施形態にかかるプロセッサのあるスレッドについての入力側FI F0蓄積量に対するスレッドの起動優先度決定方法の一例を示す図である。

【図4】

本発明の第1の実施形態にかかるプロセッサのあるスレッドについての出力側FI F0蓄積量に対するスレッドの起動優先度決定方法の一例を示す図である。

【図5】

本発明の第1の実施形態にかかるプロセッサの次に起動するスレッドを決定するためのフローの一例を示す図である。

【図6】

本発明の第2の実施形態にかかるプロセッサのあるスレッドについての入力側FI F0蓄積量に対するスレッドの起動優先度決定方法の一例を示す図である。

【図7】

本発明の第2の実施形態にかかるプロセッサのあるスレッドについての出力側FI F0蓄積量に対するスレッドの起動優先度決定方法の一例を示す図である。

**【符号の説明】**

1 0 1 FIFO(First-In First-Out)

1 0 2 選択部

1 0 3 実行制御部

1 0 4 キャッシュ

1 0 5 切替部

1 0 6 レジスタセット

1 0 7 メモリバス

1 0 8 記憶部

1 0 9 演算処理部

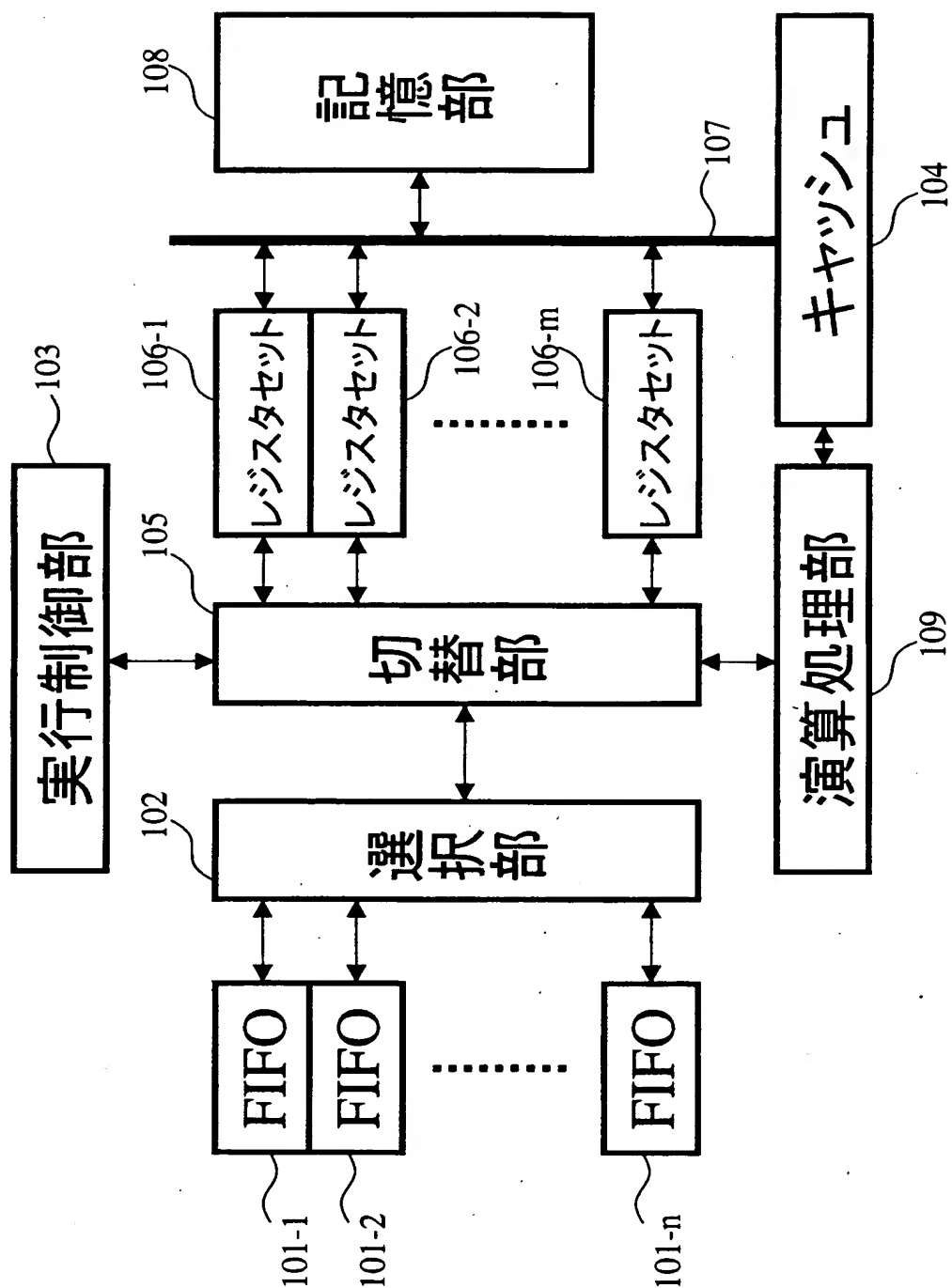
2 0 2 状態報告

2 0 3 実行指示

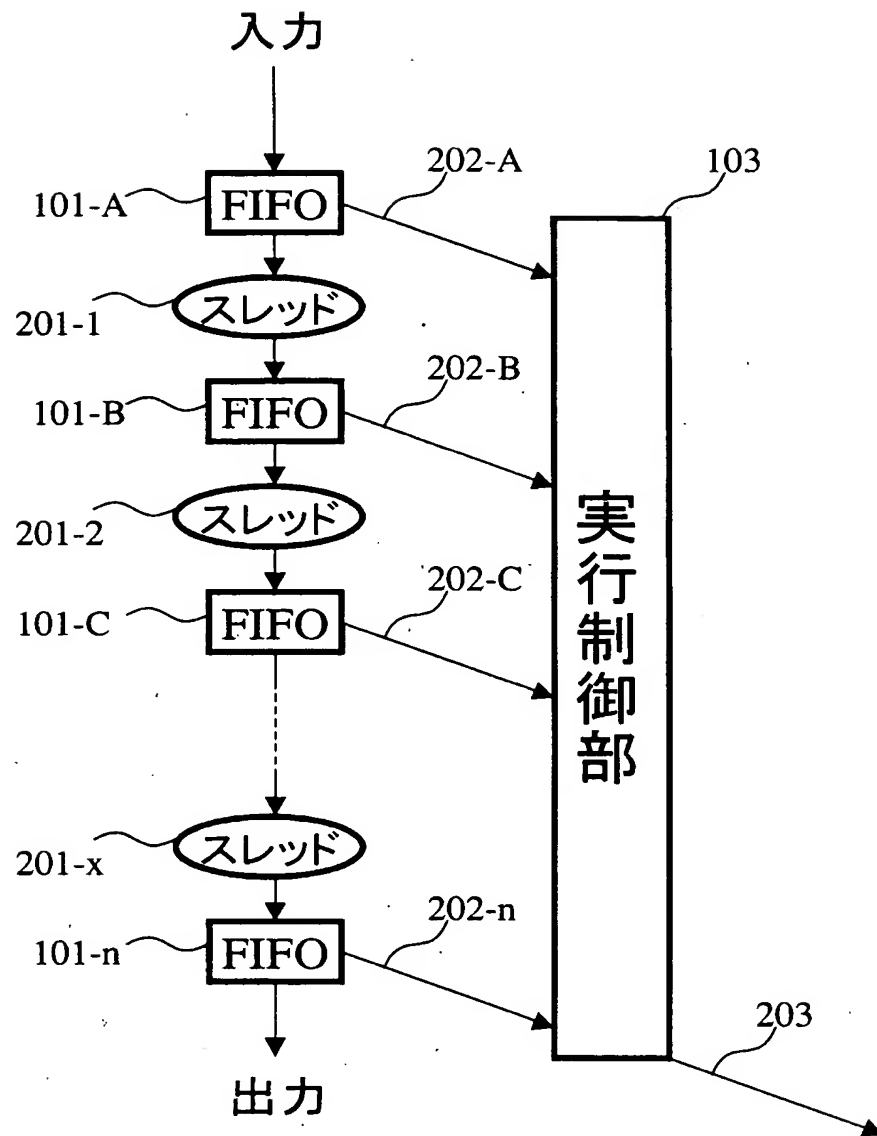
【書類名】

図面

【図 1】

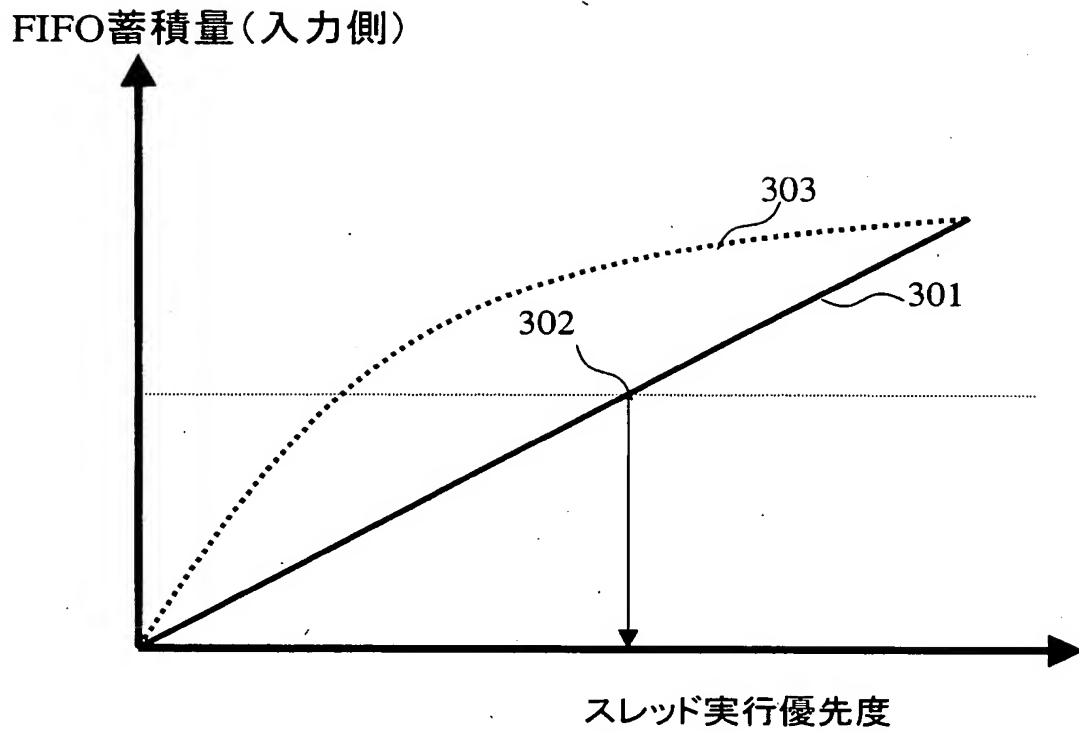


【図 2】

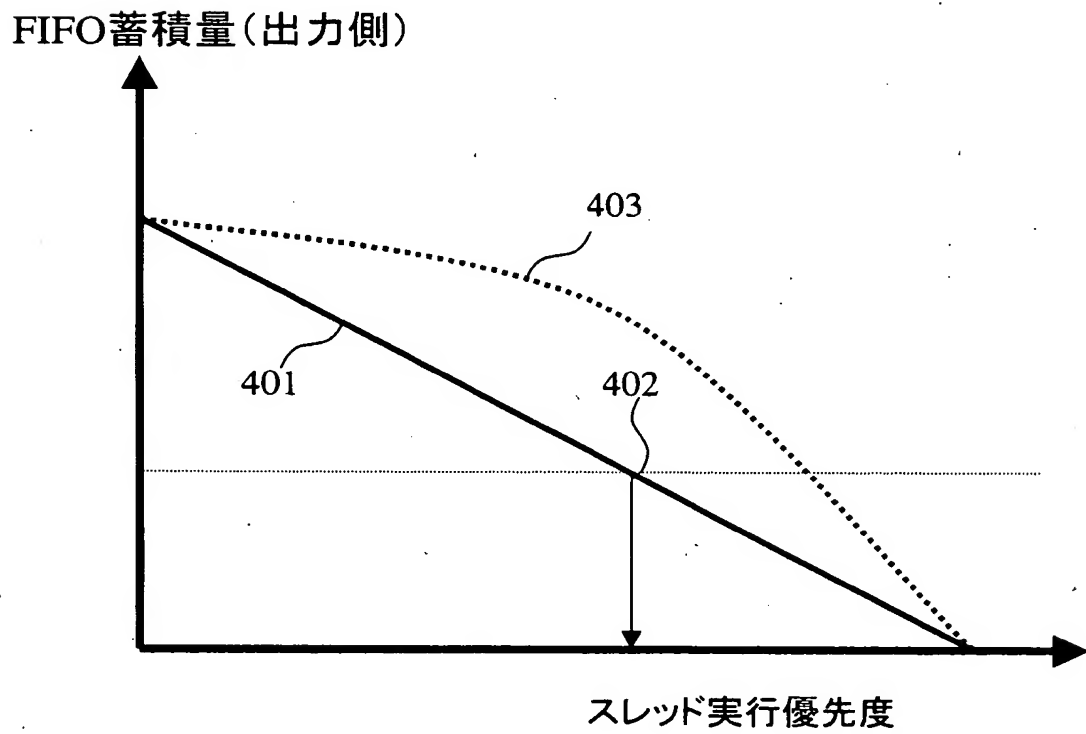




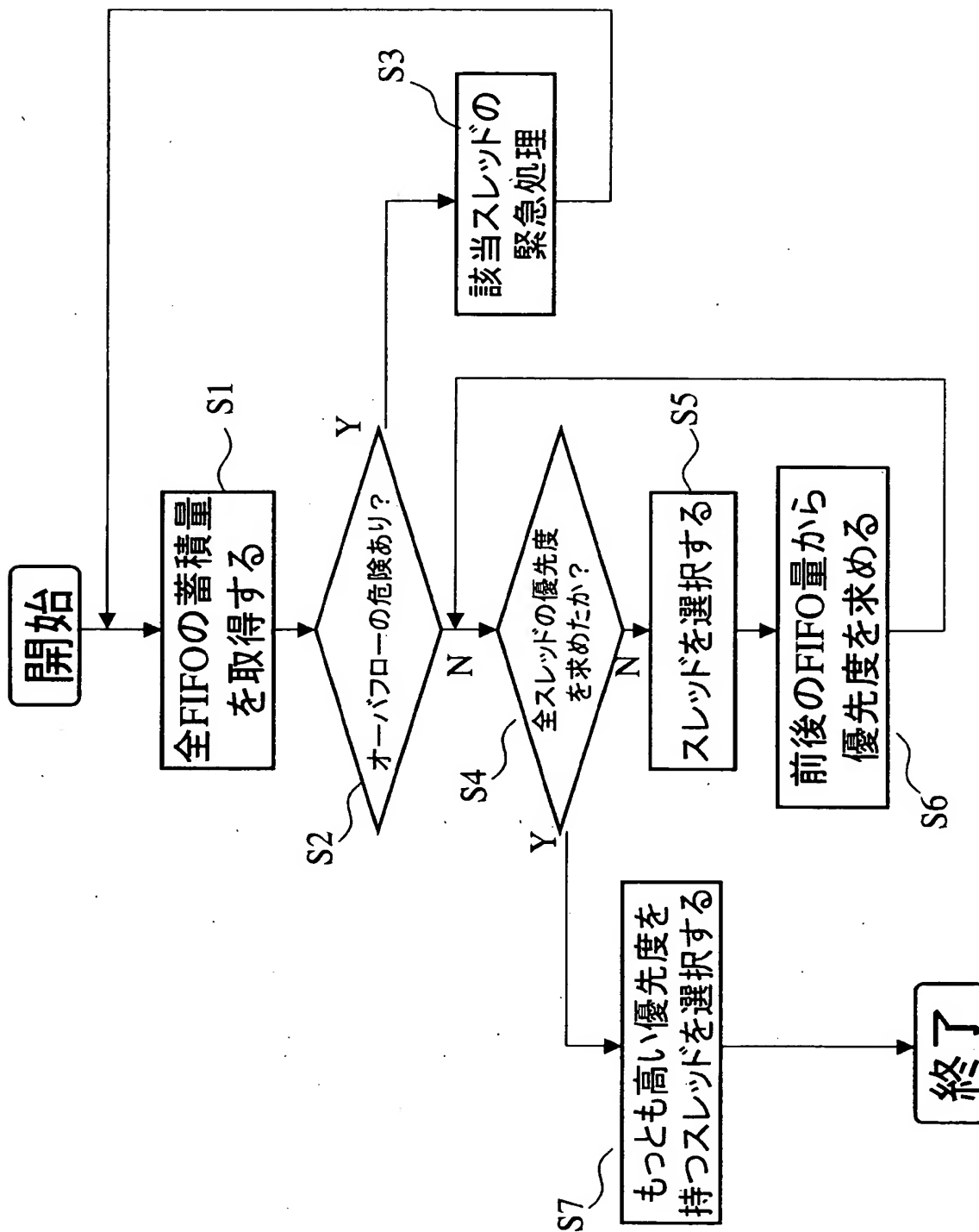
【図 3】



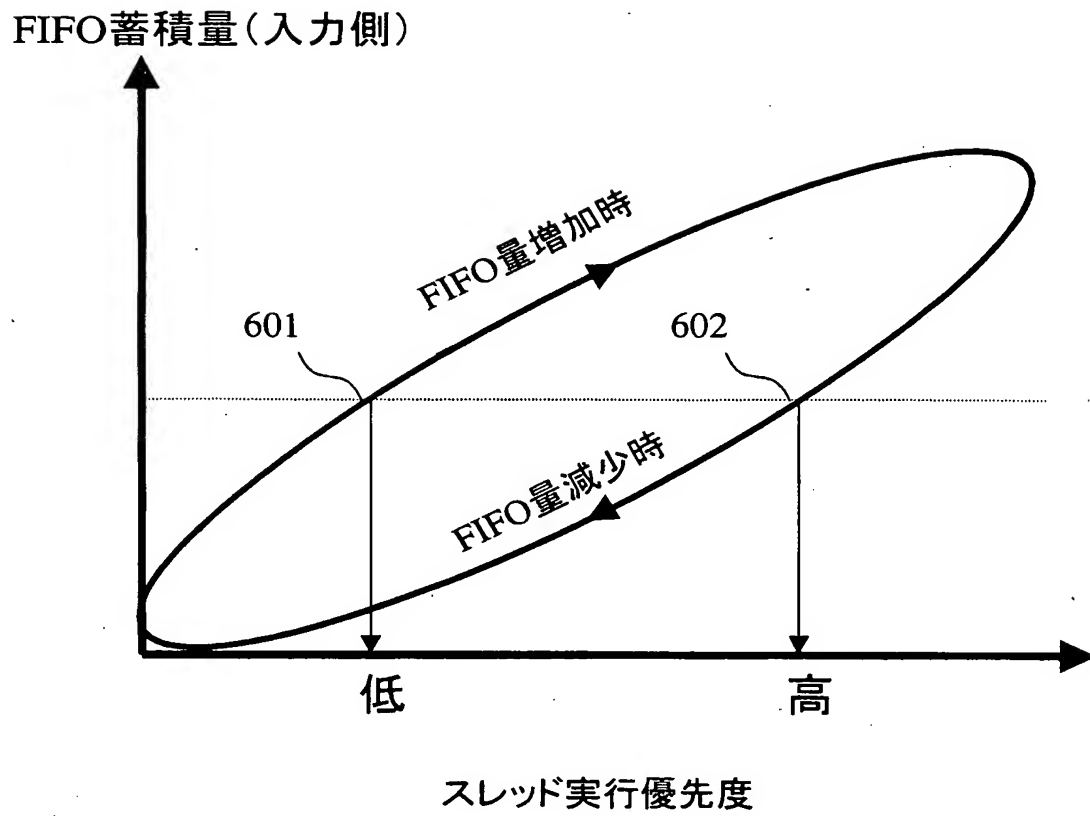
【図 4】



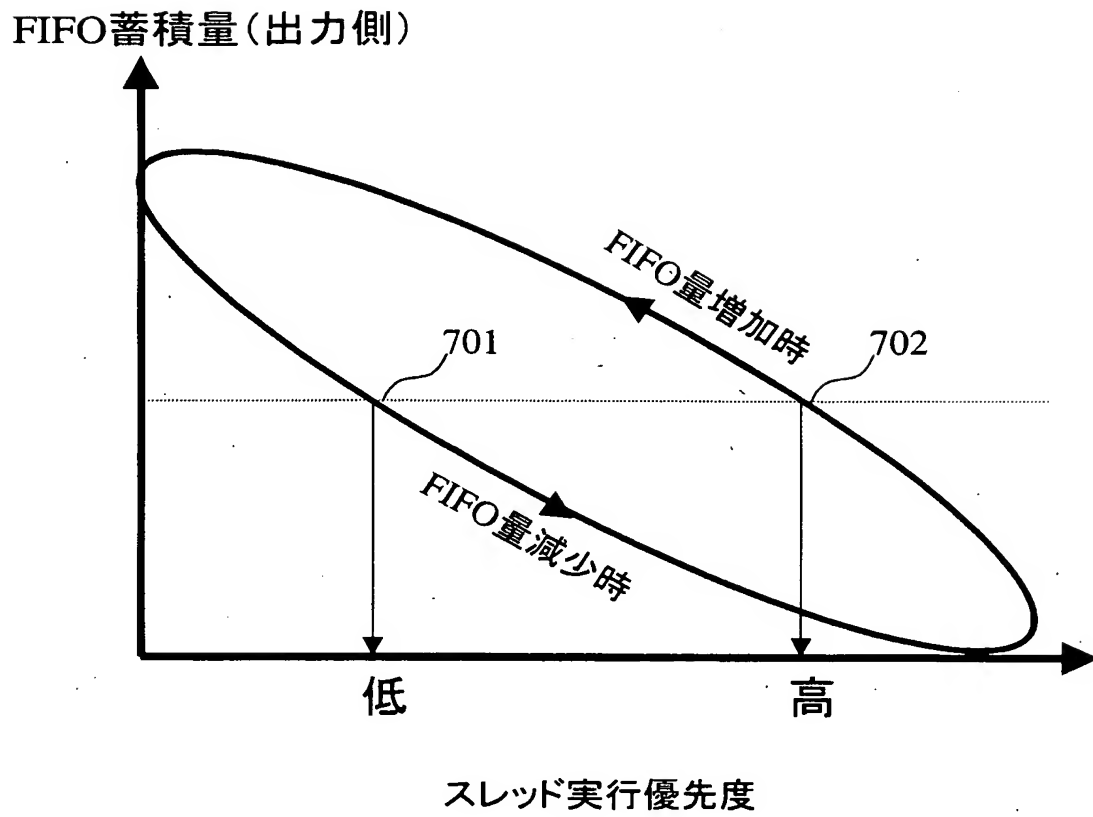
【図 5】



【図6】



【図7】



【書類名】 要約書

【要約】

【課題】 本発明はリアルタイム性の損失を最小限に抑えることが可能なデータフロー型プロセッサ、このプロセッサを利用した計算機及び起動優先度決定方法を提供することを目的とする

【解決手段】 複数の記憶手段と、時分割された処理の一つである実行単位について、前記複数の記憶手段の 1 つから入力データを受け取り、出力用記憶手段と定めた出力用記憶手段に処理した結果を記憶するデータ処理手段と、前記記憶手段のデータ記憶量の情報から実行単位の実行状況を求め、実行単位の起動優先度を決定する起動優先度決定手段とを備えたことを特徴とするプロセッサが提供される。

【選択図】 図 2

認定・付加情報

特許出願の番号	特願 2003-079478
受付番号	50300466634
書類名	特許願
担当官	第一担当上席 0090
作成日	平成15年 3月25日

<認定情報・付加情報>

【提出日】 平成15年 3月24日

次頁無

特願 2003-079478

出願人履歴情報

識別番号

[000003078]

1. 変更年月日 2001年 7月 2日  
[変更理由] 住所変更  
住 所 東京都港区芝浦一丁目1番1号  
氏 名 株式会社東芝
2. 変更年月日 2003年 5月 9日  
[変更理由] 名称変更  
住所変更  
住 所 東京都港区芝浦一丁目1番1号  
氏 名 株式会社東芝